

Разработка архитектуры адаптивных web-приложений с применением нейронных сетей

А.Д. Обухов, e-mail: obuhov.art@gmail.com

Д.В. Теселкин, e-mail: dteselk@mail.ru

ФГБОУ ВО «Тамбовский государственный технический университет»

***Аннотация.** Актуальной проблемой является персонализация интерфейсов Web-приложений в зависимости от предпочтений пользователя и особенностей его оборудования, что может повысить удобство и востребованность web-ресурса для потенциальной аудитории. Существующие математические методы и алгоритмы не всегда обеспечивают достаточной точности ввиду большой сложности данной задачи. Предлагается архитектура Web-приложений, основанная на сборе, анализе и обработке информации о пользователе и его оборудовании, а также применении нейронных сетей для выбора оптимальных параметров интерфейса. Рассмотрена практическая реализация архитектуры на Python в виде web-приложения интернет-магазина.*

***Ключевые слова:** web-приложения, персонализация интерфейса, автоматизация анализа данных, нейронные сети.*

Введение

Важными задачами при разработке программных модулей информационных систем являются обеспечение качества и эффективности их работы, однако, все большую востребованность находят такие функции, как оптимизация системы под конкретное устройство пользователя и адаптация интерфейса под его индивидуальные предпочтения для повышения комфорта использования. В предметной области web-разработки это также является актуальным вопросом [1].

Проблема разработки удобных и адаптивных интерфейсов заключается в невозможности разработчиков охватить, учесть и реализовать все субъективные пожелания каждого отдельного пользователя в частности. Из всех путей решения проблемы адаптации интерфейса под специфические характеристики и запросы пользователя самым перспективным направлением можно отметить использование методов машинного обучения [2].

Целью работы является разработка архитектуры web-приложений, позволяющей упростить решение задачи адаптации интерфейса. Для этого необходимо сформировать структуру системы, осуществить выбор необходимых инструментов и библиотек для интеграции нейронных сетей.

1. Выбор архитектуры адаптивного web-приложения

Для решения поставленной задачи предложена следующая архитектура web-приложения (рис. 1). Рассмотрим основные инструменты и технологии, используемые для ее реализации.

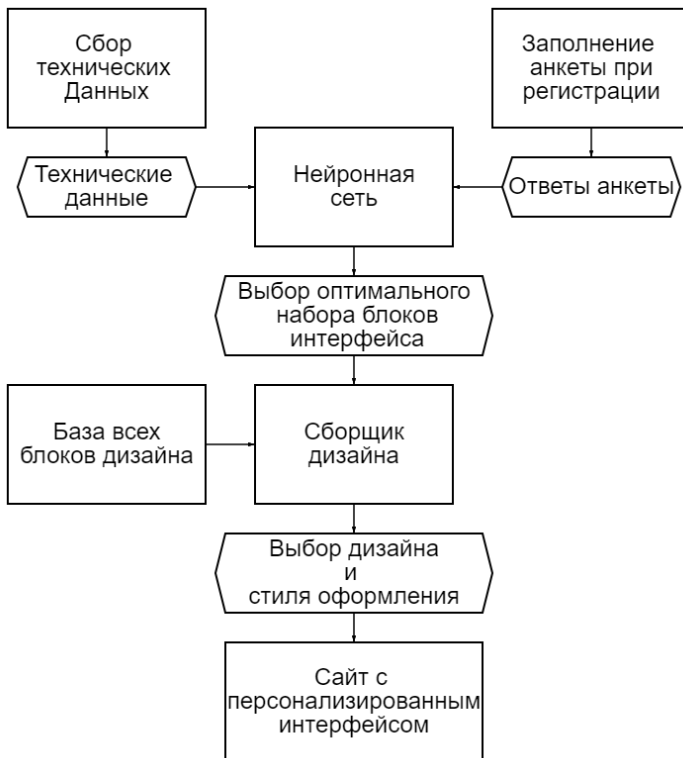


Рис. 1. Архитектура адаптивного web-приложения

При выборе языка программирования для реализации web-приложения учитывалось, что большинство современных решений прикладных задач в области машинного обучения осуществляется на

базе языка Python и таких библиотек как Keras, PyTorch и Tensorflow. В качестве основы для проведения экспериментов будем использовать Keras – библиотеку машинного обучения, главными достоинствами которой являются высокая скорость прототипирования для быстрого проведения многочисленных тестов и широкая реализация модулей для построения нейронных сетей.

Для реализации web-приложений и упрощения интеграции нейронных сетей в них возможно использование фреймворков Django и Flask. Использование их вместе с HTML и шаблонами CSS из набора инструментов Bootstrap позволит создавать адаптивные web-приложения.

Рассмотрим алгоритм реализации архитектуры адаптивного web-приложения [3]:

1. Анализируется конкретная информационная система и на основе полученных данных выбираются параметры интерфейса, которые нужно персонализировать в данной предметной области.

2. На основе полученных данных происходит выборка подходящих ключевых характеристик пользователя для данной информационной системы. Определяется степень их значимости.

3. Формируются обучающиеся данные для нейронной сети. В наборе данных X будет содержаться нормированный вектор выявленных ключевых характеристик. В свою очередь в наборе данных Y будут находиться данные о изменяемых параметрах интерфейса.

4. Формируется архитектура нейронной сети. Из-за большой вариативности и сложности данного этапа подходящая композиция слоев может быть выбрана не с первого раза. В таком случае структуру нейронной сети нужно корректировать с учетом полученных данных до тех пор, пока она не станет решать задачу с достаточной точностью.

5. Обучение нейронной сети. Полученная модель сохраняется для последующего использования и дообучения.

6. Подготовка прикладных программных интерфейсов (API) для последующей интеграции в информационную систему. Система будет обращаться к API, отправляя набор данных о пользователе, а API в свою очередь будут выдавать системе нужные параметры интерфейса.

7. Внедрение API в web-приложение. Обеспечение их взаимодействия – передачи данных о пользователе в API и применение ответных параметров для адаптации интерфейса.

8. Работа адаптации интерфейса. При первом заходе на сайт система собирает данные о пользовательском устройстве и выдает ему нейтральный интерфейс, эффективно работающий при полученных характеристиках устройства пользователя. При регистрации

пользователь вносит личные данные, они передаются в API, которые выдают на их основе настройки интерфейса, оптимальные для пользователей с данным набором признаков. Параметры сохраняются в базе данных и выставляются по умолчанию для данного пользователя.

9. Если пользователь захочет изменить интерфейс, он заходит в настройки и изменяет необходимые ему параметры. Это информация сохраняется в базе данных для последующего дообучения нейронной сети при накоплении значимого объема данных для увеличения ее точности и поддержки актуальности.

Представленная архитектура адаптивных web-приложений и алгоритм ее реализации будет апробирован на примере тестового интернет-магазина.

2. Реализация адаптивного web-приложения для предметной области Интернет-торговли

На первом этапе реализации необходимо осуществить обучение нейронной сети на основе анализа предметной области и информации о входных и выходных параметрах.

Рассмотрим модель нейронной сети для подбора оптимальных параметров интерфейса под конкретного пользователя (рис. 2). На вход поступает нормализованный вектор данных о пользователе из 30 значений. Он проходит через плотный слой из 256 нейронов. Применяется слой прореживания для снижения эффекта переобучения. После чего используется еще 3 плотных слоя по 512, 256 и 9 нейронов. Последний слой является выходным, соответствующим 9 выходным параметрам интерфейса.

Обученная модель показала точность в размере 79%, для обучения использовалось около 2000 записей, полученных в ходе опроса сотрудников и студентов университета, а также путем размещения опросного листа в web-форме.

Далее осуществлено сравнение полученного решения, основанного на применении нейронных сетей, с алгоритмическим решением, основанном на статистическом анализе собранных данных.

Нейронные сети за счет анализа пользовательских данных и предпочтений показывает немного большую точность, которая в дальнейшем с ростом количества обрабатываемых данных может быть улучшена. В текущем эксперименте нейронная сеть показывает на 5-10% лучшую точность на контрольных замерах. С ростом объема исходных данных точность может быть повышена.

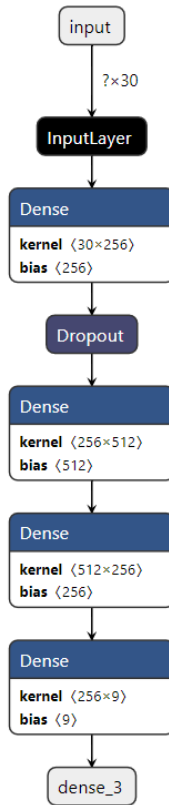


Рис. 2. Модель нейронной сети

Применение нейронных сетей позволило значительно снизить сложность программного кода (по метрике Холстеда) на 20%. Большой эффект оказывает тот факт, что разработчику не требуется анализировать и выявлять закономерности в исходных данных, так как эту работу принимает на себя нейронная сеть.

Сложность по метрике Джилба за счет значительного сокращения ветвлений и циклов сократилась в 1.5 раза.

Сокращение сложности реализации привело к тому, что при оценке реализации тестового проекта методом СОСОМО II трудоемкость и время стоимость реализации значительно сокращается: на 11% и 36% соответственно.

Рассмотрим процесс функционирования тестового web-приложения.

Пользователь входит на сайт. Посредством библиотек JavaScript собираются данные о его устройстве. Данные передаются в API адаптации на сервере, который нормализует их в бинарный вектор и передает в обученную нейронную сеть.

Полученные данные о необходимых параметрах интерфейса устанавливаются, как подходящие для временного пользователя.

При регистрации пользователя он заполняет анкету с данными о себе. Данные также передаются в API адаптации на сервере для обработки в нейронной сети.

Полученные данные о нужных параметрах интерфейса сохраняются в базе данных и устанавливаются, как подходящие для конкретного пользователя. При повторном заходе этого же пользователя в систему параметры интерфейса загружаются из базы данных.

Пример работы тестового web-приложения представлен на рис. 3 и рис. 4 для двух различных типов пользователей.

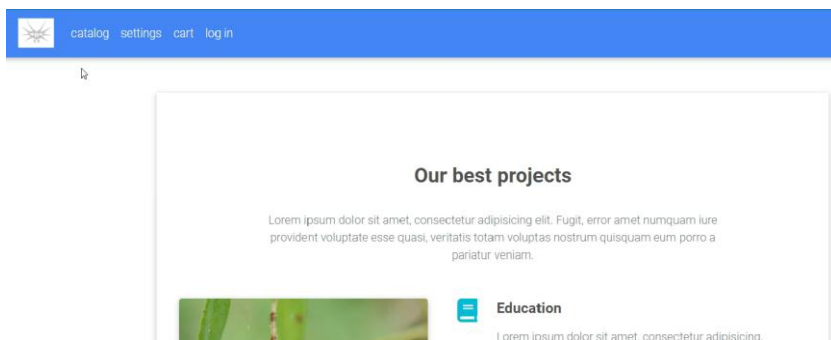


Рис. 3. Пример адаптивного web-приложения для Интернет-торговли для первого пользователя

Первый тип представлен студентом, при анкетировании отметившим, что предпочитает получать доступ к максимуму информации, а анализ его оборудования показал, что он входит на сайт с ноутбука с небольшой диагональю. Это привело к выбору такой схемы интерфейса, при которой отображается полное меню и весь контент, но сам дизайн не перегружен лишними блоками, чтобы обеспечить оптимальную нагрузку на оборудование пользователя.

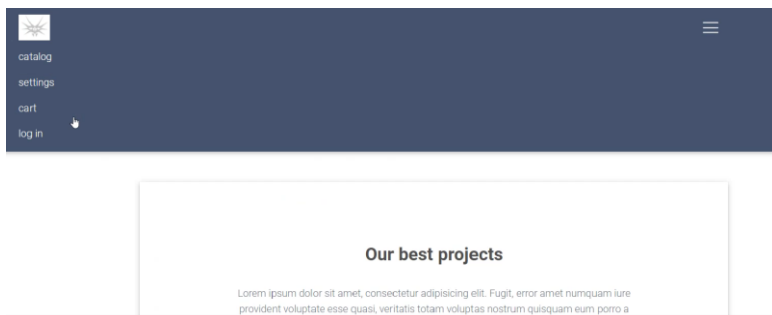


Рис. 4. Пример адаптивного web-приложения для Интернет-торговли для второго пользователя

Второй тип пользователя относится к старшей возрастной группе, а оценка уровня производительности его оборудования показала, что он заходит на сайт с маломощного стационарного компьютера. Для удобства пользователя выбран минималистичный дизайн с минимумом графических элементов, меню может скрываться, чтобы сконцентрировать внимание на главной странице с основными разделами.

Таким образом, получено решение задачи адаптации интерфейса web-приложения под индивидуальные особенности пользователя и характеристики его устройства.

Дальнейшие исследования будут связаны с реализацией возможности адаптации компоновки блоков разметки web-страницы для большей вариативности интерфейса.

Заключение

Сформулирован подход к выбору и реализации архитектуры web-приложения, направленный на интеграцию информационной системы с нейронными сетями для автоматизации подбора персональных настроек интерфейса. Это позволит автоматизировать процесс адаптации системы под предпочтения пользователя и особенности его устройства (тип, мощность, размеры экрана и так далее). Использование нейронных сетей по сравнению с модулями, основанными на статистическом анализе, может обеспечить большую точность.

Полученные результаты могут использоваться при реализации адаптивных web-приложений в различных предметных областях и сферах деятельности.

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации, грант Президента РФ МК-74.2020.9

Литература

1. Ерхов Р. В. Методы разработки и архитектура современных веб-приложений //Новые информационные технологии в научных исследованиях. – 2016. – С. 129-130.

2. Зосимов В. В., Христоворов А. В., Булгакова А. С. Программные решения для динамического изменения пользовательского интерфейса на основе автоматически собранной информации о пользователе //Труды Института системного программирования РАН. – 2018. – Т. 30. – №. 3. – С. 207-220.

3. Обухов А.Д., Краснянский М.Н., Николюкин М.С. Нейросетевой метод адаптации параметров интерфейса информационных систем / Вестник Воронежского государственного технического университета. – 2020. – Т. 16. – №. 5. – С. 26-31